# **Trial 3 and final:**

## **Requirements Document for Emotion Classification Model**

**1. Project Goals**

* Develop a machine learning model to classify emotions from Twitter messages.
* Achieve high performance in terms of accuracy, precision, recall, and F1-score.
* Ensure the model is robust and generalizes well to unseen data.
* Evaluate the model using relevant metrics and visualizations.

**2. Functional Requirements**

* **Data Preprocessing**: Clean and preprocess the text data (e.g., tokenisation, stop-word removal, stemming/lemmatisation).
* **Feature Extraction**: Convert text data into numerical features using techniques like TF-IDF or word embeddings.
* **Model Training**: Train the model using the preprocessed data.
* **Model Evaluation**: Evaluate the model using metrics such as accuracy, precision, recall, F1-score, and confusion matrix.
* **Model Validation**: Use cross-validation to ensure the model's robustness.
* **Model Testing**: Test the model on a separate test dataset to evaluate its performance on unseen data.
* **Visualisation**: Generate graphs to visualise the model's performance (e.g., ROC curve, precision-recall curve).

**3. Non-Functional Requirements**

* **Performance**: The model should achieve high accuracy and other relevant metrics.
* **Scalability**: The solution should be scalable to handle larger datasets in the future.
* **Usability**: The process should be well-documented and easy to understand.
* **Maintainability**: The code should be modular and maintainable for future updates.
* **Latency**: The model should make predictions in real-time or near real-time.

**4. Technical Stack**

* **Programming Language**: Python
* **Libraries**:
* Data Processing: Pandas, NumPy
* Text Processing: NLTK, SpaCy
* Feature Extraction: Scikit-learn, Gensim
* Model Building: Scikit-learn, TensorFlow/Keras, PyTorch
* Evaluation: Scikit-learn, Matplotlib, Seaborn
* **Development Environment**: Jupyter Notebook or any Python IDE
* **Version Control**: Git

**5. Data Sources**

* **Primary Data Source**: Emotions Kaggle dataset (Twitter messages with emotion labels)
* **Data Format**: CSV file with columns for text and corresponding emotion label

**6. Expected Behaviour**

* **Input**: A CSV file containing Twitter messages and their corresponding emotion labels.
* **Output**: A trained machine learning model capable of classifying emotions in new Twitter messages.
* **User Interaction**: Users will provide the dataset and run the training script. The model will output performance metrics and visualisations.

**7. Input-Output Details**

* **Input**:
* CSV file with columns: text (Twitter message) and emotion (label: 0-5)
* **Output**:
* Trained model file (e.g., .h5, .pkl)
* Performance metrics (accuracy, precision, recall, F1-score)
* Visualisations (confusion matrix, ROC curve, precision-recall curve)

**8. Performance Metrics**

* **Accuracy**: The ratio of correctly predicted instances to the total instances.
* **Precision**: The ratio of correctly predicted positive observations to the total predicted positives.
* **Recall**: The ratio of correctly predicted positive observations to the all observations in actual class.
* **F1-Score**: The weighted average of Precision and Recall.
* **Confusion Matrix**: A table used to describe the performance of a classification model.
* **ROC Curve**: A graph showing the performance of a classification model at all classification thresholds.
* **Precision-Recall Curve**: A graph showing the trade-off between and recall for different thresholds.

**9. Constraints**

* **Latency**: The model should be able to make predictions in real-time or near real-time.
* **Deployment Requirements**: The model should be deployable on a local machine with the specified technical stack.
* **Hardware Requirements**: Sufficient computational resources (CPU/GPU) to handle model training and inference.

**10. Additional Considerations**

* **Data Privacy**: Ensure that the data used complies with relevant data privacy regulations.
* **Ethical Considerations**: Avoid biases in the model by ensuring a balanced dataset and fair evaluation metrics.

This requirements document outlines the key aspects of developing a high-performance emotion classification model using the Kaggle dataset. It includes project goals, functional and non-functional requirements, technical stack, data sources, expected behaviour, input-output details, performance metrics, and constraints.

## **User Stories and Tasks for Emotion Classification Model**

### Using the Agile Story assistant: Giving it the generated Requirements document

**Requirement summary**: Develop a machine learning model to classify emotions from Twitter messages.

**User Story 1**

**Content**: As a data scientist, I want to clean and preprocess the text data so that it is suitable for model training.

**Acceptance Criteria**:

* The text data should be tokenised.
* Stop-words should be removed from the text data.
* Stemming or lemmatisation should be applied to the text data.
* The preprocessing steps should be documented and reproducible.

**Tasks**:

1. Tokenise the text data.

* **Goal**: Split the text into individual tokens.
* **Expected Outcome**: Text data is tokenised.
* **Completion Criteria**: Tokenised text data is saved and documented.

1. Remove stop-words from the text data.

* **Goal**: Eliminate common words that do not contribute to the model.
* **Expected Outcome**: Stop-words are removed.
* **Completion Criteria**: Cleaned text data is saved and documented.

1. Apply stemming or lemmatisation.

* **Goal**: Reduce words to their base or root form.
* **Expected Outcome**: Text data is stemmed or lemmatised.
* **Completion Criteria**: Processed text data is saved and documented.

**User Story 2**

**Content**: As a data scientist, I want to convert the preprocessed text data into numerical features using techniques like TF-IDF or word embeddings so that the model can understand the data.

**Acceptance Criteria**:

* The text data should be converted into numerical features using TF-IDF or word embeddings.
* The feature extraction process should be documented and reproducible.
* The numerical features should be stored in a format suitable for model training.

**Tasks**:

1. Implement TF-IDF feature extraction.

* **Goal**: Convert text data into TF-IDF features.
* **Expected Outcome**: Text data is represented as TF-IDF features.
* **Completion Criteria**: TF-IDF features are saved and documented.

1. Implement word embeddings feature extraction.

* **Goal**: Convert text data into word embeddings.
* **Expected Outcome**: Text data is represented as word embeddings.
* **Completion Criteria**: Word embeddings are saved and documented.

**User Story 3**

**Content**: As a data scientist, I want to train the machine learning model using the preprocessed and feature-extracted data so that it can classify emotions from Twitter messages.

**Acceptance Criteria**:

* The model should be trained using the preprocessed and feature-extracted data.
* The training process should be documented and reproducible.
* The model should be saved in a format suitable for future use (e.g., .h5, .pkl).

**Tasks**:

1. Select and implement the machine learning algorithm.

* **Goal**: Choose an appropriate algorithm for emotion classification.
* **Expected Outcome**: Algorithm is selected and implemented.
* **Completion Criteria**: Algorithm implementation is documented.

1. Train the model using the preprocessed data.

* **Goal**: Train the model with the prepared dataset.
* **Expected Outcome**: Model is trained.
* **Completion Criteria**: Trained model is saved and documented.

**User Story 4**

**Content**: As a data scientist, I want to evaluate the model using metrics such as accuracy, precision, recall, F1-score, and confusion matrix so that I can assess its performance.

**Acceptance Criteria**:

* The model should be evaluated using accuracy, precision, recall, F1-score, and confusion matrix.
* The evaluation results should be documented and reproducible.
* The evaluation metrics should be visualised using appropriate graphs (e.g., ROC curve, precision-recall curve).

**Tasks**:

1. Calculate evaluation metrics.

* **Goal**: Compute accuracy, precision, recall, F1-score, and confusion matrix.
* **Expected Outcome**: Evaluation metrics are calculated.
* **Completion Criteria**: Metrics are documented.

1. Generate visualisations for evaluation metrics.

* **Goal**: Create graphs for ROC curve, precision-recall curve, and confusion matrix.
* **Expected Outcome**: Visualisations are generated.
* **Completion Criteria**: Visualisations are included in the report.

**User Story 5**

**Content**: As a data scientist, I want to validate the model using cross-validation so that I can ensure its robustness.

**Acceptance Criteria**:

* The model should be validated using cross-validation.
* The cross-validation process should be documented and reproducible.
* The cross-validation results should be included in the model evaluation report.

**Tasks**:

1. Implement cross-validation.

* **Goal**: Validate the model using cross-validation.
* **Expected Outcome**: Cross-validation is performed.
* **Completion Criteria**: Cross-validation results are documented.

**User Story 6**

**Content**: As a data scientist, I want to test the model on a separate test dataset so that I can evaluate its performance on unseen data.

**Acceptance Criteria**:

* The model should be tested on a separate test dataset.
* The test dataset should be different from the training and validation datasets.
* The test results should be documented and reproducible.

**Tasks**:

1. Prepare the test dataset.

* **Goal**: Ensure the test dataset is ready for evaluation.
* **Expected Outcome**: Test dataset is prepared.
* **Completion Criteria**: Test dataset is documented.

1. Evaluate the model on the test dataset.

* **Goal**: Test the model's performance on unseen data.
* **Expected Outcome**: Model is evaluated on the test dataset.
* **Completion Criteria**: Test results are documented.

**User Story 7**

**Content**: As a data scientist, I want to generate visualisations to show the model's performance so that stakeholders can easily understand the results.

**Acceptance Criteria**:

* Visualisations should include a confusion matrix, ROC curve, and precision-recall curve.
* The visualisations should be clear and easy to interpret.
* The visualisations should be included in the model evaluation report.

**Tasks**:

1. Create a confusion matrix visualisation.

* **Goal**: Visualise the confusion matrix.
* **Expected Outcome**: Confusion matrix is visualised.
* **Completion Criteria**: Confusion matrix is included in the report.

1. Create an ROC curve visualisation.

* **Goal**: Visualise the ROC curve.
* **Expected Outcome**: ROC curve is visualised.
* **Completion Criteria**: ROC curve is included in the report.

1. Create a precision-recall curve visualisation.

* **Goal**: Visualise the precision-recall curve.
* **Expected Outcome**: Precision-recall curve is visualised.
* **Completion Criteria**: Precision-recall curve is included in the report.

**User Story 8**

**Content**: As a user, I want to provide a CSV file containing Twitter messages and their corresponding emotion labels so that the model can be trained.

**Acceptance Criteria**:

* The input CSV file should contain columns for text (Twitter message) and emotion (label: 0-5).
* The CSV file should be correctly formatted and free of errors.
* The model training script should accept the CSV file as input.

**Tasks**:

1. Validate the input CSV file format.

* **Goal**: Ensure the CSV file is correctly formatted.
* **Expected Outcome**: CSV file format is validated.
* **Completion Criteria**: Validated CSV file is documented.

1. Implement CSV file input handling in the training script.

* **Goal**: Ensure the training script can read the CSV file.
* **Expected Outcome**: CSV file is read by the script.
* **Completion Criteria**: Input handling is documented.

**User Story 9**

**Content**: As a user, I want to receive a trained machine learning model capable of classifying emotions in new Twitter messages so that I can use it for predictions.

**Acceptance Criteria**:

* The trained model should be saved in a format suitable for future use (e.g., .h5, .pkl).
* The model should be capable of making predictions in real-time or near real-time.
* The model should be deployable on a local machine with the specified technical stack.

**Tasks**:

1. Save the trained model in a suitable format.

* **Goal**: Ensure the model is saved for future use.
* **Expected Outcome**: Model is saved in .h5 or .pkl format.
* **Completion Criteria**: Saved model is documented.

1. Test the model's prediction latency.

* **Goal**: Ensure the model can make predictions in real-time or near real-time.
* **Expected Outcome**: Model's prediction latency is tested.
* **Completion Criteria**: Latency results are documented.

1. Ensure the model is deployable on a local machine.

* **Goal**: Verify the model can be deployed with the specified technical stack.
* **Expected Outcome**: Model is deployable.
* **Completion Criteria**: Deployment process is documented.

**User Story 10**

**Content**: As a developer, I want the code to be modular and maintainable so that future updates can be easily implemented.

**Acceptance Criteria**:

* The code should be organised into modules with clear responsibilities.
* The code should be well-documented and follow best practices.
* The code should be version-controlled using Git.

Tasks:

Refactor the code into modular components.

Goal: Ensure the code is organised into modules with clear responsibilities.

Expected Outcome: Code is modular.

Completion Criteria: Modular code is documented.

Document the codebase.

Goal: Provide clear documentation for the code.

Expected Outcome: Code is well-documented.

Completion Criteria: Documentation is complete and accessible.

Implement version control using Git.

Goal: Ensure the code is version-controlled.

Expected Outcome: Code is tracked using Git.

Completion Criteria: Git repository is set up and maintained.

User Story 11

Content: As a stakeholder, I want to ensure that the data used complies with relevant data privacy regulations so that the project adheres to ethical standards.

Acceptance Criteria:

The data should be anonymised where necessary.

The data usage should comply with relevant data privacy regulations.

The data privacy measures should be documented and reviewed.

Tasks:

Anonymise the data where necessary.

Goal: Ensure that any sensitive information is anonymised.

Expected Outcome: Data is anonymised.

Completion Criteria: Anonymised data is documented.

Review data privacy regulations.

Goal: Ensure compliance with relevant data privacy regulations.

Expected Outcome: Data usage complies with regulations.

Completion Criteria: Compliance is documented.

Document data privacy measures.

Goal: Provide clear documentation of data privacy measures.

Expected Outcome: Data privacy measures are documented.

Completion Criteria: Documentation is complete and accessible.

Jira Board Setup

To put these tasks into a Jira board, follow these steps:

Create a new project in Jira for the Emotion Classification Model.

Create Epics for each user story.

Example Epics: Data Preprocessing, Feature Extraction, Model Training, Model Evaluation, Model Validation, Model Testing, Visualisation, User Input Handling, Model Deployment, Code Maintainability, Data Privacy.

Create Stories under each Epic based on the user stories provided.

Create Tasks under each Story based on the tasks provided.

Assign tasks to team members and set due dates as appropriate.

Track progress by moving tasks through the Jira workflow stages (To Do, In Progress, Done).

This setup will help ensure that all aspects of the project are covered and that tasks are clearly defined and trackable.